Dan begin annotating the web page in real-time like a shared whiteboard (**87**). As they discuss a few changes, Bob is asking Dan to integrate a project logo into the web page. Dan agrees but is pressured now to run to another meeting. He says good-bye to Bob and tells him that he will check with him next day. Dan closes all his windows and as he leaves, his name turns gray throughout all of his shared objects displayed on Bob's client.

[**0044**] Now alone, Bob continues annotating the web page. He also types in a few lines for Dan in the chat window before closing it. He then right clicks on the chat object and creates a new shared file object. He picks the logo file from his local file system and the file object becomes part of Bob's and Dan's activity thread (**88**). Bob closes all windows and leaves. Next morning when Dan returns to his office, he finds Bob's additional annotations, his chat message, and the project logo file. He starts working on the web page and few hours later, he puts the reworked page into the activity thread as a shared file object (**89**) and adds a message with some comments (**90**). He also shares these two objects with Catherine (**91**) so that she can download and deploy the newly revised web page and logo.

[**0045**] This scenario demonstrates how the collaboration server enables a user to move seamlessly and effortlessly back and forth from private to public information and from asynchronous to synchronous real-time collaboration, without manually creating a shared workspace or setting up a conference. Collaboration starts off with a single shared object and evolves into a multi-object activity, which is structured by a dynamic group of participants as they create and add new shared objects. An activity thread provides the conversational context and awareness for an emerging collaboration; it allows aggregating a mix of different object types. The various objects (message, chat, file etc.) presented in this scenario could be represented through GSOs managed by the collaboration server and the hierarchical relationship of activity threads could be managed through the relation database of the collaboration server.

[**0046**] Thus a collaboration server and operation thereof has been described with regard to various figures. As mentioned above, the functional components of the collaboration server and client API may be implemented in any combination of hardware and software. **FIGS. 4 and 5** are state diagrams and flowchart illustrations of methods, apparatus (systems) and computer program products according to an embodiment of the invention. It will be understood that each block of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These computer program instructions may be loaded onto a computer or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the computer or other programmable data processing apparatus create means for implementing the functions specified in the flowchart block or blocks. These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[**0047**] Those skilled in the art should readily appreciate that programs defining the functions of the present invention can be delivered to a computer in many forms; including, but not limited to: (a) information permanently stored on non-writable storage media (e.g. read only memory devices within a computer such as ROM or CD-ROM disks readable by a computer I/O attachment); (b) information alterably stored on writable storage media (e.g. floppy disks and hard drives); or (c) information conveyed to a computer through communication media for example using baseband signaling or broadband signaling techniques, including carrier wave signaling techniques, such as over computer or telephone networks via a modem.

[**0048**] While the invention is described through the above exemplary embodiments, it will be understood by those of ordinary skill in the art that modification to and variation of the illustrated embodiments may be made without departing from the inventive concepts herein disclosed. Moreover, while the preferred embodiments are described in connection with various illustrative program command structures, one skilled in the art will recognize that the system may be embodied using a variety of specific command structures. Accordingly, the invention should not be viewed as limited except by the scope and spirit of the appended claims.

What we claim is:

1. A server comprising:

a storage device for storing an object data structure having a plurality of entries defining attributes of an object, the plurality of attributes apportioned into general entries and variable entries, the general entries including a list of members having access to the object and the variable entries including at least one item of persistent information associated with the object; and

broadcast logic for broadcasting changes to the object data structure to the at least one member having access to the object.

2. The server of claim 1, wherein the storage device stores a plurality of object data structures associated with a plurality of objects, each one of the plurality of object data structures including a plurality of entries defining attributes of a corresponding one of the plurality of objects, the plurality of entries including a list of members having access to the associated object, the server further including:

a navigation logic including a relation database identifying relationships between different ones of the plurality of objects, the navigation logic also including logic for identifying a set of the plurality of objects that are affected by a change to a data structure of a given object of the plurality of objects; and

wherein, the broadcasting logic further includes means for broadcasting a change to a data structure of the given object to members in the list of members of each object in the set of the plurality of objects identified by the navigation logic.